**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously presented) A method in a data processing system for allocating memory by a memory allocation function, comprising:

receiving a memory request for a reference to a block of memory;

returning the reference to the block of memory to satisfy the request;

forming a plurality of linked-lists referring to memory blocks of a plurality of sizes, each of the plurality of linked-lists referring to memory blocks of a common size;

setting a fast access tree to refer to a first one of the plurality of linked-lists; and

setting a general access tree to refer to a second one and a third one of the plurality of linked-lists, wherein a size of a memory block referred to by the first linked-list is larger than a size of a memory block referred to by the second linked-list and smaller than a size of a memory block referred to by the third linked-list.

2. (Previously presented) The method of claim 1, further comprising adjusting a structure of the fast access tree and the general access tree based on the memory request.

3. (Previously presented) The method of claim 2, further comprising searching the fast access tree to satisfy the memory request and, if the memory request is not satisfied, searching the general access tree to satisfy the memory request.

4. (Previously presented) The method of claim 1, further including ensuring that the fast access tree refers to one of the plurality of linked-lists that is most frequently requested.

5. (Canceled)

6. (Withdrawn) A method in a data processing system for providing access to a memory that includes an operating system with a system memory call, the memory further including a program which includes a memory access function, comprising the steps performed by the memory access function of:

requesting access to a portion of memory via the system memory call;

receiving from the system memory call a pointer to the portion of memory;

dividing the portion of memory into memory blocks, a plurality of the memory blocks being of different sizes;

forming a plurality of linked-lists, each linked-list referring to memory blocks of a common size, each linked-list having an associated counter;

setting a fast access tree to refer to a first of the plurality of linked-lists;

setting a general access tree to refer to a second of the plurality of linked-lists;

receiving a memory request;

determining which among the plurality of linked-lists contains a memory block that will satisfy the memory request;

incrementing the counter associated with the determined linked-list;

returning a reference to the memory block on the determined linked-list;

comparing the counters of the plurality of linked-lists to identify a predetermined number of linked-lists with a largest counter; and

ensuring that the fast access tree is set to refer to the identified linked-lists with the largest counter.

7. (Previously presented) A system for allocating memory, comprising:

means for receiving a memory request for a reference to a block of memory;

means for returning the reference to the block of memory to satisfy the request; and

means for forming a plurality of linked-lists referring to memory blocks of a plurality of sizes, each of the plurality of linked-lists referring to memory blocks of a common size;

means for setting a fast access tree to refer to a first one of the plurality of linked-lists;

means for setting a general access tree to refer to a second one and a third one of the plurality of linked-lists, wherein a size of a memory block referred to by the first linked-list is larger than a size of a memory block referred to by the second linked-list and smaller than a size of a memory block referred to by the third linked-list.

8. (Previously presented) A data processing system for providing access to memory, comprising:

a memory including:

an access tree structure comprising a fast access tree and a general access tree;

a program including a memory access function that provides access to the memory, forms a plurality of linked-lists referring to memory blocks of a plurality of sizes, each of the plurality of linked-lists referring to memory blocks of a common size, sets the fast access tree to refer to a first one of the plurality of linked-lists, and sets the general access tree to refer to a second one and a third one of the plurality of linked-lists, wherein a size of a memory block referred to by the first linked-list is larger than a size of a memory block referred to by the second linked-list and smaller than a size of a memory block referred to by the third linked-list; and

a processor for executing the program.

9. (Previously presented) The data processing system of claim 8, further including an operating system with a system memory function, and wherein the memory access function provides access to the memory by utilizing the system memory function.

10. (Previously presented) The data processing system of claim 8, wherein the memory access function adjusts a structure of the fast access tree and the general access tree based on a memory request.

11. (Previously presented) The data processing system of claim 8, wherein the fast access tree refers to one of the plurality of linked-lists that is most frequently accessed.

12. (Previously presented) The data processing system of claim 8, wherein a most frequently accessed memory block size is included in the fast access tree.

13. (Canceled)

14. (Previously presented) The data processing system of claim 8, wherein a least frequently accessed memory block size is included in the general access tree.

15. (Previously presented) The data processing system of claim 8, wherein the memory access function searches the fast access tree to satisfy a memory request and, if the memory request is not satisfied, searches the general access tree to satisfy the memory request .

16. (Original) The data processing system of claim 15, wherein each of the plurality of linked-lists has an associated counter indicating a number of times that the associated linked-list has been accessed.

17. (Previously presented) A computer-readable medium including instructions for performing a method for allocating memory by a memory allocation function, the method comprising:

receiving a memory request for a reference to a block of memory;

returning the reference to the block of memory to satisfy the request;

forming a plurality of linked-lists referring to memory blocks of a plurality of sizes, each of the plurality of linked-lists referring to memory blocks of a common size;

setting a fast access tree to refer to a first one of the plurality of linked-lists;

setting a general access tree to refer to a second one and a third one of the plurality of linked-lists, wherein a size of a memory block referred to by the first linked-list is larger than a size of a memory block referred to by the second linked-list and smaller than a size of a memory block referred to by the third linked-list.

18. (Currently amended) The computer-readable medium, including instructions for performing the method, of claim 17, the method further comprising adjusting a structure of the fast access tree and the general access tree based on the memory request.

19. (Currently amended) The computer-readable medium, including instructions for performing the method, of claim 17, the method further comprising searching the fast access tree to satisfy the memory request and, if the memory request is not satisfied, searching the general access tree to satisfy the memory request.

20. (Canceled)

21. (Currently amended) The computer-readable medium, including instructions for performing the method, of claim 17, the method further comprising ensuring that the fast access tree refers to one of the plurality of linked-lists that is most frequently requested.

22. (Canceled)

BLANK PAGE

23. (Canceled)

24. (Previously presented) The method of claim 1, wherein receiving, returning, forming, and setting are performed in a user space of the memory.

25. (Previously presented) A method in a data processing system for allocating memory, the method comprising:

forming a plurality of linked-lists with associated references to memory blocks;

setting a fast access tree to refer to a first one of the linked-lists;

setting a general access tree to refer to a second one and a third one of the linked-lists, wherein any memory blocks referred to by any references associated with the first linked-list are larger in size than any memory blocks referred to by any references associated with the second linked-list and smaller in size than any memory blocks referred to by any references associated with the third linked-list;

receiving a memory request for a reference to a memory block; and

accessing at least one of the fast access tree and the general access tree to return the reference to the memory block in response to the memory request.

26. (Previously presented) The method of claim 25, further comprising adjusting a structure of the fast access tree and the general access tree based on the memory request.

27. (Previously presented) The method of claim 25, further comprising searching the fast access tree to satisfy the memory request and, if the memory request is not satisfied, searching the general access tree to satisfy the memory request.

28. (Previously presented) The method of claim 25, further including ensuring that the fast access tree refers to one of the plurality of linked-lists that is most frequently requested.

29. (Previously presented) A method in a data processing system for allocating memory, the method comprising:

receiving a memory request for a reference to a memory block;

returning the reference to the memory block to satisfy the request; and

adjusting an access tree structure based on the memory request,

wherein the memory block that satisfies the request is determined based on size with reference to a plurality of linked-lists with associated references to memory blocks of a plurality of sizes, wherein any references associated with a first linked-list refer memory blocks larger than any memory blocks referred to by any references associated with a second linked-list and smaller in size than any memory blocks referred to by any references associated with a third linked-list, and

wherein the adjustment to the access tree structure reflects a result of the determination of the memory block that satisfies the memory request.

30. (Previously presented) The method of claim 29, further comprising adjusting a structure of a fast access tree and a general access tree based on the memory request.

31. (Previously presented) The method of claim 30, further comprising searching the fast access tree to satisfy the memory request and, if the memory request is not satisfied, searching the general access tree to satisfy the memory request.

32. (Previously presented) The method of claim 31, further including ensuring that the fast access tree refers to one of the plurality of linked-lists that is most frequently requested.